Journal of Information Security and Applications 48 (2019) 102363

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

A technique for DNA cryptography based on dynamic mechanisms

Md. Rafiul Biswas[a], Kazi Md. Rokibul Alam[a,*], Shinsuke Tamura[b], Yasuhiko Morimoto[c]

[a] Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh
[b] Graduate School of Engineering, University of Fukui, Fukui 910-8507, Japan
[c] Graduate School of Information Engineering, Hiroshima University, Higashi-Hiroshima 739-8521, Japan

ARTICLE INFO

Article history:
Available online xxx

Keywords:
DNA cryptography
Dynamic sequence table
Dynamic DNA encoding
NIST
Asymmetric cryptosystem

ABSTRACT

This paper proposes a technique for DNA cryptography based on dynamic mechanisms *i.e.* 'dynamic sequence table' and 'dynamic DNA encoding'. To form dynamic sequence table: initially, DNA base sequences are assigned to 256 ASCII characters randomly. Where to attain dynamism, positions of DNA base sequences are re-arranged iteratively following a mathematical series. Likewise, to form dynamic DNA encoding: the use of NCBI bank genome sequence with a mathematical series dynamically settles the number of DNA bases required to merge the ciphertext of every two chunks. Here, the way of encryption is: to transform the plaintext into DNA bases, to assign them *w.r.t.* ASCII characters using dynamic sequence table, to divide these data into a finite number of chunks, to encrypt them using an asymmetric cryptosystem, and lastly to merge the ciphertext of chunks through dynamic DNA encoding. Thus, the usage of dynamic mechanisms along with an asymmetric cryptosystem obviously enhances the secrecy level of data. Finally, the outcome of the proposed technique, a comparative study with existing techniques, the security analyses, and a statistical test according to the National Institute of Standards and Technology (NIST) to analyze the randomness of the generated ciphertext are presented.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

While encrypting data, DNA cryptography aims to ensure enhanced secrecy than traditional cryptography through combining biological as well as computational characteristics [1]. Usually, traditional cryptosystems offer only single fold security, and possibly, their secrecy is breached while the underlying computational techniques are revealed. In contrast, through self-assembling properties of DNA bases [2] along with a cryptographic technique, DNA cryptography provides multi fold securities [3] that enrich the secrecy level of data.

Existing DNA cryptographic techniques extensively deploy symmetric cryptosystems whereas only a few others employ asymmetric cryptosystems. In addition, often they treat DNA bases statically. Symmetric cryptosystems offer the same key regularly for both encryption and decryption purposes [4], which degrade their effectiveness. Whereas in asymmetric cryptosystems, the role of different keys for encryption and decryption purposes, and their underlying efficient mathematical concepts [5] ensure improved usability. Hence, to encrypt data; the usage of DNA bases with the art of dynamism along with a modern cryptosystem is definitely advantageous.

To achieve these requirements, this paper develops new security components *i.e.* dynamic sequence table [6] and dynamic DNA encoding [7] to propose a technique for DNA cryptography. Here, the data encryption mainly consists of three stages *i.e.* the generation of dynamic sequence table, the usage of an asymmetric cryptosystem and the formation of dynamic DNA encoding. At first, dynamic sequence table assigns ASCII characters for DNA base sequences distinctly. Then the generated output is split into a number of fixed sized chunks to be encrypted by an asymmetric cryptosystem. Finally, dynamic DNA encoding settles the number of DNA bases from NCBI bank database [29] to merge every two contiguous chunks. Thus, the proposed technique assures enhanced secrecy for any form of data *e.g.* text, image, audio, numeric, video etc.

The rest of this paper is arranged as follows: Section 2 reviews related works. Section 3 describes the proposed technique. Section 4 explains the experimental analysis, Section 5 presents an analysis of security and statistics, and finally, Section 6 concludes the paper.

* Corresponding author.
E-mail addresses: rafiulbiswas@gmail.com, rokib@cse.kuet.ac.bd (Md.R. Biswas), rokib@cse.kuet.ac.bd (K.Md.R. Alam), tamura@dance.plala.or.jp (S. Tamura), morimo@hiroshima-u.ac.jp (Y. Morimoto).

https://doi.org/10.1016/j.jisa.2019.102363
2214-2126/© 2019 Elsevier Ltd. All rights reserved.

2. Related works

Recently a number of DNA cryptographic techniques have been proposed. To encrypt the data, techniques proposed in [8–10] mainly exploits one-time pad (OTP) along with DNA bases. Here the technique proposed in [8] stores the ciphertext in microdot and uses PCR technology while decryption. An alike technique proposed in [9] is assisted by DNA addition, DNA subtraction etc., and uses a Feistel network. The technique proposed in [10] uses PCR based biological technology.

Techniques proposed in [11–14] are mainly based on DNA steganography. Using it, the technique proposed in [11] stores the key and the ciphertext. To encrypt the plaintext, the technique proposed in [12] deploys DNA based play fair algorithm and substitution technique to increase the confusion level. The hybrid technique proposed in [13] represents the plaintext as matrix form using RGB colors and encapsulates the key by DNA steganography. The technique proposed in [14] relies on secure medium and chemical reactions.

The technique proposed in [15] relies on NCBI bank genome sequence [29] to generate the key and develops DNA Vigenere cipher to enrich the secrecy level. Another technique proposed in [16] initially complements the binary value for each ASCII character. Then it transforms the binary value into DNA bases and finally shifts them from MSB to LSB. Using the concept of Diffie and Hellman, the technique proposed in [17] generates the shared key. Then, it applies a hybridization technique to produce the ciphertext, though herein, there is no need to transfer the key.

The technique proposed in [18] re-arranges DNA bases according to the look-up table. Another technique proposed in [19] applies sixteen rounds of encryption on the plaintext. The technique proposed in [20] exploits DNA storage, encrypts through the random private key and swaps the ciphertext in parallel to form small clusters. To ensure more secrecy, the technique proposed in [21] adopts multi-level actions that deploy the selection of the round key, the transformation of ciphertext, and the shift operation. To enhance the strength of traditional DES, the technique proposed in [30] integrates DNA cryptography to increase its keyspace and permutation.

Techniques proposed in [31–33] deal with the encryption of image. While encrypting a bunch of images, the technique proposed in [31] adopts DNA sequence operations and image matrix indexes. Here the use of DNA sequence with cellular automata enriches its secrecy. Alongside, the exploitation of index-based permutation and diffusion makes its process faster. The technique proposed in [32] exploits a hyper-chaotic system in gyrator transform domains to deepen the secrecy of color image. In the cryptosystem, the random sequences of the chaotic system act as primary keys. Again, the parameters of the chaotic system and the transform behave as extra keys. The multi-stage encryption technique proposed in [33] repeatedly employs visual cryptography, steganography and one-time-pad to maintain both the secrecy and the quality of an image.

The technique proposed in [22] initially employs the genetic algorithm (GA) and Needleman-Wunsch algorithm to produce the key. Then the encryption and the decryption are performed through DNA computing *i.e.* by applying different biological operations namely, transcription, translation, DNA sequencing, deep learning tool etc. Another alike hybrid technique proposed in [23] deploys DNA sequence along with the GA to encrypt the image. Here at first, GA minimizes the correlation of adjacent pixels of the image. Then XOR operation on the pixel value produces DNA sub-strings that is used as the key to encrypt the image.

Analyzing above techniques it is seen that a huge number of existing works often combine static DNA bases with machine learning tools, symmetric cryptosystems like OTP, DES etc. Although a few techniques consider asymmetric cryptosystems, rarely they handle DNA bases dynamically. However, the usage of dynamic characteristics of DNA bases along with a modern cryptosystem significantly can improve the secrecy level of data. Hence exploiting dynamic mechanisms with an asymmetric cryptosystem, this paper proposes a new technique for DNA cryptography.

3. The proposed technique

The proposed technique consists of four distinct stages *i.e.* the generation of dynamic sequence table, the formation of dynamic DNA encoding, the encryption process and the decryption process, which are described below.

3.1. Dynamic sequence table [6]

Dynamic sequence table assigns DNA base sequences for 256 ASCII characters dynamically. To generate the dynamic sequence table, at first 256 ASCII characters are mapped randomly with 256 distinct DNA base sequences alike to [24]. Where each DNA base sequence consists of four DNA bases (*i.e.* A, T, G, and C) which are assigned as 'A' = 00, 'T' = 01, 'G' = 10 and 'C' = 11. Prior to data transmission, no one except the sender and the receiver only share the initial status of this table *i.e.* Table 1. Now the sender finitely iterates the positions of DNA base sequences according to a mathematical series *e.g.* the Fibonacci series. In addition, only the involved parties know the series in advance. Here, only the positions of DNA base sequences are re-arranged at every iteration. But, positions of ASCII characters remain unchanged. The sender executes the process as follows.

Step 1: Generates the Fibonacci series as $F_{(i)} = F_{(i\text{-}1)} + F_{(i\text{-}2)}$ where $F_{(0)} = 1$, and $F_{(1)} = 1$.
Step 2: Picks an arbitrary value from the series. Uses this value for the first iteration.
Step 3: Scans the first character of the plaintext; picks the corresponding ASCII value of it.
Step 4: Checks the value of Step 3; it is either odd or even. If odd, sets the variable 'Bool' as Bool = 1. Otherwise sets Bool = 0.
Step 5: If Bool = 1, re-arranges all DNA base sequences forwardly according to (Item Number + the value of Fibonacci series)% 256. If Bool = 0, re-arranges all DNA base sequences backwardly according to the value of a variable 'Result'. Where, Result = (Item Number − the value of Fibonacci series)% 256. If Result < = 0, sets Result = Result + 256.
Step 6: Uses the next consecutive values (either in ascending or descending order) of the Fibonacci series for the next iterations.

Table 1
Initial status of dynamic sequence table.

Item number	DNA bases	ASCII character
1	AAAA	x
2	AATA	Y
3	AAGA	p
4	AACA	?
5	CCAA	=
6	GTTT	{
7	TATA	+
8	CCGC	9
...	...	...
...	...	...
254	AGGC	M
255	GGGG	I
256	CACA	*

Step 7: Encrypts 'the initial arbitrary value chosen from the series', 'the required iteration numbers', and 'the value of Bool' and sends them to the receiver after the encryption process (will be discussed in Section 3.3).

Prior to decryption of the ciphertext, the receiver needs to decrypt the parameters of Step 7 to iterate the table equal to the sender.

3.2. Dynamic DNA encoding [7]

Dynamic DNA encoding combines ciphertext of every two contiguous chunks dynamically. To develop it, a mathematical series *e.g.* Fibonacci series is used, which is only shared by both the sender and the receiver. Besides, only the involved parties maintain the NCBI bank genome sequence in advance. The selection of an arbitrary value from the Fibonacci series to pick DNA base sequences from NCBI bank offers dynamism. The sender executes the process as follows.

Step 1: Picks an arbitrary value j from the Fibonacci series to choose the equal numbers of DNA base sequences m from NCBI bank database *i.e.* m_j refers the chosen DNA base sequences for the jth position of Fibonacci series.
Step 2: Employs the chosen DNA base sequences to merge the ciphertext of chunks. If there are N chunks, to generate the final ciphertext they are merged as final ciphertext $= \sum_{i=1}^{N} Cipher\ of\ Chunk(i) + \sum_{j=1}^{N} chosen\ DNA\ bases(m_j)$, where it picks next consecutive values (either in ascending or descending order) of Fibonacci series to merge next every two contiguous chunks. Thus, the repetition of the process merges all chunks.
Step 3: Encrypts 'the initial arbitrary value chosen from the series' to send it to the receiver after the encryption process.

Prior to decryption of the ciphertext, the receiver decrypts the parameter of Step 3. While decryption, it removes the extra (because of merging) DNA base sequences from ciphertext of every two contiguous chunks.

3.3. Encryption process

At first, the receiver generates keys of the chosen asymmetric cryptosystem, announces the public key to be used by the sender to encrypt the plaintext and stores the private key for itself to be used for decryption. Now the sender encrypts the plaintext as follows. Fig. 1 depicts the encryption process.

Step 1: Initially scans the plaintext and converts each character into its corresponding 3−digit ASCII value.
Step 2: Transforms each integer digit into its' corresponding 4-digit binary value.
Step 3: Converts the binary value into DNA bases as 'A' = 00, 'T' = 01, 'G' = 10 and 'C' = 11, same as Section 3.1.
Step 4: Uses dynamic sequence table (*i.e.* the final form, as discussed in Section 3.1) to assign the ASCII character for each DNA base sequence.
Step 5: Transforms each ASCII character to its' corresponding integer value.
Step 6: Divides it into a finite number of fixed sized chunks. If required, adds the necessary number of 0′s at the left of the integer value of the first chunk.
Step 7: Encrypts each chunk using an asymmetric cryptosystem.
Step 8: Transforms ciphertext of each chunk into DNA bases following Step 2 and Step 3.
Step 9: Merges every two contiguous chunks using dynamic DNA encoding. This is the final ciphertext.

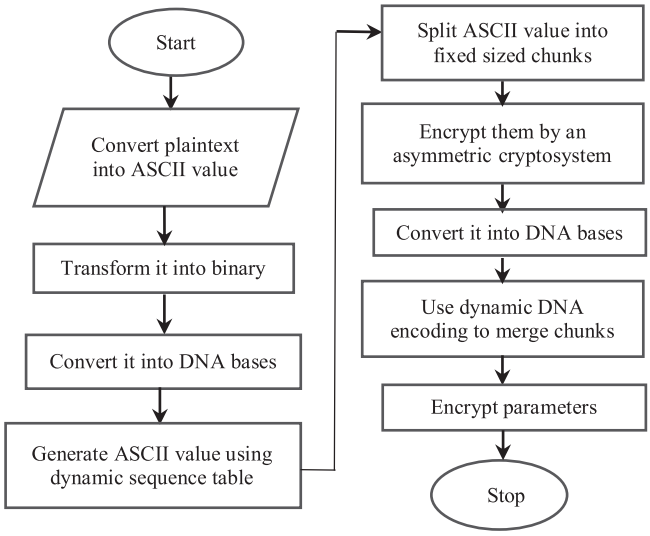

Start
Convert plaintext into ASCII value
Transform it into binary
Convert it into DNA bases
Generate ASCII value using dynamic sequence table
Split ASCII value into fixed sized chunks
Encrypt them by an asymmetric cryptosystem
Convert it into DNA bases
Use dynamic DNA encoding to merge chunks
Encrypt parameters
Stop

Fig. 1. Encryption process.

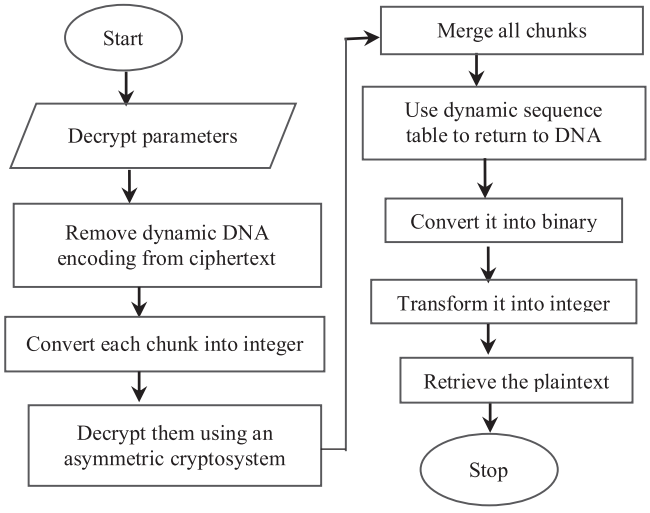


Fig. 2. Decryption process.

Step 10: At last, sends the size of the chunk, and parameters used to generate the dynamic sequence table along with those of dynamic DNA encoding, in encrypted form.

3.4. Decryption process

Already the receiver possesses 'the initial form of dynamic sequence table' (*i.e.* Table I), 'the mathematical series', and 'the DNA sequences of NCBI bank'. Through decryption, it retrieves the plaintext. Although it is the reverse process of the encryption process, the receiver executes it as follows. Fig. 2 depicts the process.

Step 1: At first, decrypts parameters received from the sender.
Step 2: Removes the dynamic DNA encoding portion from the ciphertext. Thus, the exact ciphertext of chunks is obtained.
Step 3: Converts the ciphertext of every chunk into binary form as 'A' = 00, 'T' = 01, 'G' = 10, 'C' = 11 and then transforms every 4-digit binary into corresponding single digit integer value.
Step 4: Decrypts every chunk using an asymmetric cryptosystem.
Step 5: Merges the chunks altogether. If exists, discards extra 0′s from the left of the integer value of the first chunk.

Table 2
Output of dynamic sequence table.

Step	Operation
Step 1	Fibonacci series: 1, 1, 2, 3, 5, 8, 13, 21, 34. . . . .
Step 2	Pick the value 2 from the series for the 1st iteration
Step 3	1st character of the plaintext is W (ASCII value = 87)
Step 4	Set 'Bool'= 1 as 87 is an odd value
Step 5	After 1st iteration, Table 1 becomes as: (A portion is shown)
	Item number / DNA bases / ASCII character: 1 / GGGG / x; 2 / CACA / Y; — / — / —; 256 / AGGC / M
Step 6	Next value of the series is 3 and it is used for 2nd iteration
Step 7	Encrypt values: '2', '1', '1' for the receiver

Table 3
Output of dynamic DNA encoding.

Step	Operation
Step 1	Pick the value 5 from the series Pick 1st 5 DNA bases from NCBI database *i.e.* **CACCT**
Step 2	(*GACGGA)**CACCT**(*TCCATA)**ACCTCGCA**(*—) —.
Step 3	Encrypt the data 5 for the receiver

* Only a portion of data is shown.

Step 6: Transforms into the corresponding ASCII characters.
Step 7: Uses the final dynamic sequence table to match with the corresponding DNA base sequences.
Step 8: Converts DNA base sequences into binary form as 'A' = 00, 'T' = 01, 'C' = 11, 'G' = 10.
Step 9: Transforms each 4-digit binary value into its corresponding single digit integer value.
Step 10: Considers each 3-digit integer value to transform its' corresponding ASCII character, and retrieves the plaintext.

4. Experimental analysis

4.1. Experimental setup

To develop the prototype of the proposed technique, asymmetric cryptosystems namely, RSA, ElGamal, and Paillier are chosen separately while considering 2048 bit key size. It is developed under the environment of Intel Core i5-8250 U 1.60–3.40 GHz 64-bit processor with 8 GBytes of RAM running on Windows 10 operating system. For coding, programming language Java with version jdk1.8.1 is used along with Eclipse IDE. Considering the plaintext "Welcome to CSE, KUET" along with 2048 bit RSA key size, the output of the proposed technique is presented below.

4.2. Output of dynamic sequence table

For the stage of dynamic sequence table of Section 3.1, the output is presented in Table 2.

4.3. Output of dynamic DNA encoding

For the stage of dynamic DNA encoding of Section 3.2, the output is presented in Table 3.

4.4. Output of encryption process

For the stage of the encryption process of Section 3.3, the output is presented in Table 4.

4.5. Output of decryption process

For the stage of the decryption process of Section 3.4, the output is presented in Table 5.

4.6. Experimental results and discussions

Considering different data sizes of plaintexts along with different cryptosystems as already mentioned, the size of the corresponding ciphertexts for the proposed technique are depicted in Fig. 3. In various plaintexts (*i.e.* the number of test), the numbers of characters are 18,954, 27,632 and 34,219, respectively. From the figure, it is observed that for every asymmetric cryptosystem, the length of the ciphertext is many times larger than the length of the plaintext. Similarly considering the same plaintexts along with the same cryptosystems, the time requirements of encryption and decryption processes for the proposed technique are presented in Fig. 4. From the figure, it is seen that for RSA, the time requirement of decryption is greater than that of encryption. The reason is: both RSA encryption and decryption functions exploit modular exponentiation, and after encryption, the size of the ciphertext becomes larger than the size of the plaintext. However, for ElGamal and Paillier, the encryption time is greater than that of decryption.

Table 4
Output of encryption process.

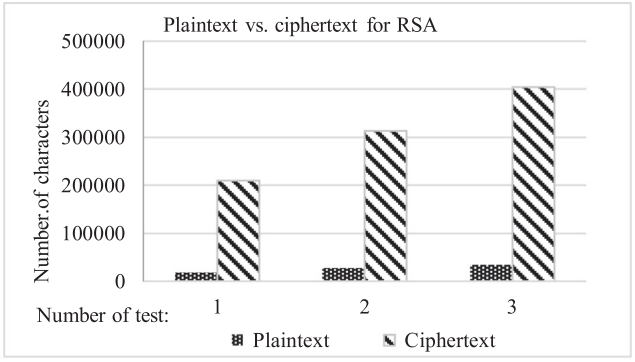
Step	Operation	
Step 1	Plaintext	Welcome to CSE, KUET
	ASCII value	
Step 2	Binary value*	
Step 3	DNA Base*	AAGATTAAGTCCCCTACTCGCC
Step 4	Uses dynamic sequence table *i.e. Step* 5 of Table 2	
	ASCII Character*	dHG%FsD@bj&VM
Step 5	Integer value*	
Step 6	# of chunk = 3, # of data in every chunk = 20	
	Chunk 1* = 0,000,010,007; Chunk 2* = 5,068,064,098; —	
Step 7	Cipher (chunk 1*)	
	Cipher (chunk 2*)	
Step 8	DNA bases (chunk 1*)	CTTATTGCCTTACGATATTGCGACCGACCCTTAAAACGACCGAAG
	DNA bases (chunk 2*)	TATAAAGGTTATCGACCGATAAAAAAAAACCTTATTGATTGAAAG
Step 9	Using dynamic DNA encoding merges ciphertext of chunks.	
	Final Ciphertext*	CTTATTG**ATTGA**TATAAAGGTT**CCGAGAGA**ACCTTAC
Step 10	Encrypt values: '20', and others, already shown in Step 7 of Table 2 and Step 3 of Table 3.	

* Only a portion of data is shown.

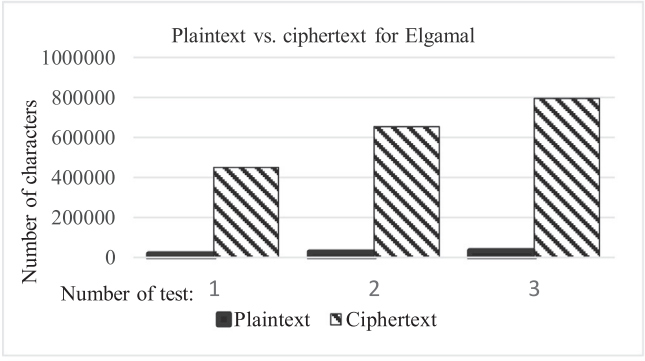
Table 5
Output of decryption process.

Step	Operation	
Step 1	Retrieves values of parameters *i.e.* '20', '2', '1', '1', '5'.	
Step 2	Removes DNA bases of dynamic DNA encoding	
	Cipher (chunk 1*)	CTTATTGCCTTACGATATTGCGAC
	Cipher (chunk 2*)	TATAAAGGTTATCGACCGATAAGT
Step 3	Binary (chunk 1*)	
	Binary (chunk 2*)	
	Integer (chunk 1*)	
	Integer (chunk 2*)	
Step 4	Decrypted (chunk 1*)	
	Decrypted (chunk 2*)	
Step 5	Merged chunk*	
Step 6	ASCII form*	dHG%FsD@bj&VM
Step 7	Uses dynamic sequence table *i.e. Step* 5 of Table 2	
	DNA base*	GACCTGGAGTGTCCTACTCGCCTC
Step 8	Binary value	
Step 9	3−digit ASCII value*	
Step 10	Retrieved Plaintext	Welcome to CSE, KUET

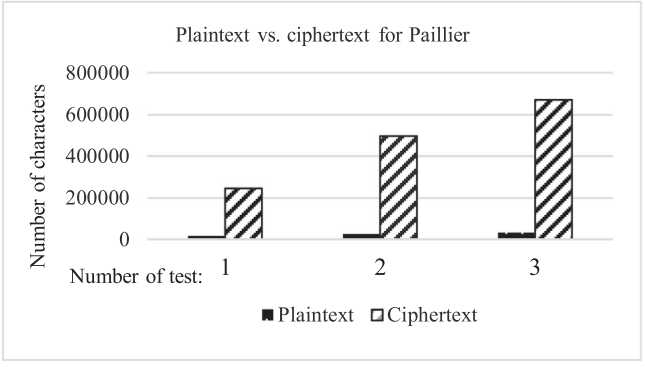
* Only a portion of data is shown.



(a)

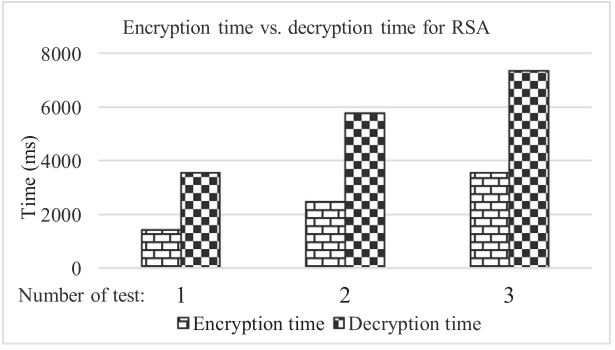


(b)

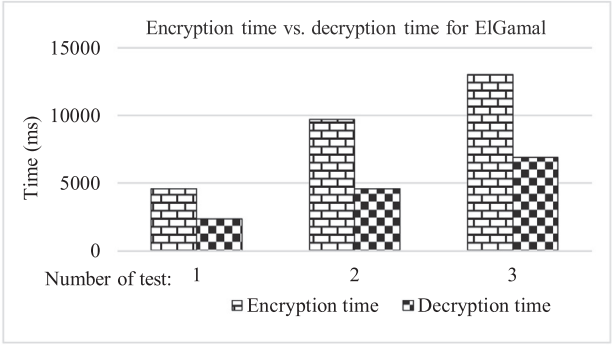


(c)

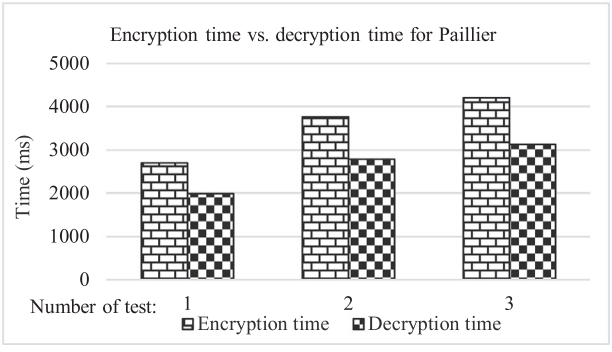
Fig. 3. For the proposed technique plaintext vs. ciphertext with different data sizes while considering: (a) RSA, (b) ElGamal, and (c) Paillier.



(a)



(b)



(c)

Fig. 4. For the proposed technique encryption time vs. decryption time with different data sizes while considering: (a) RSA, (b) ElGamal, and (c) Paillier.

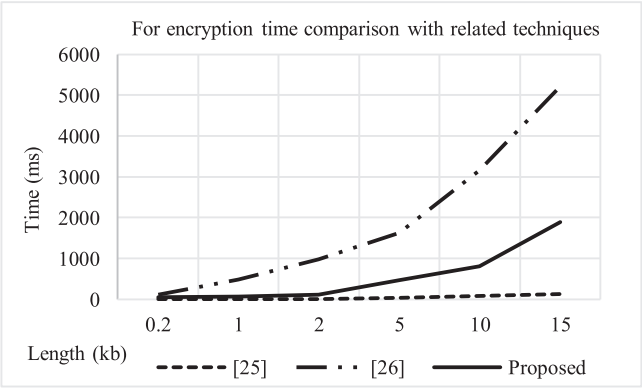


Fig. 5. Comparison in case of encryption time with related techniques.

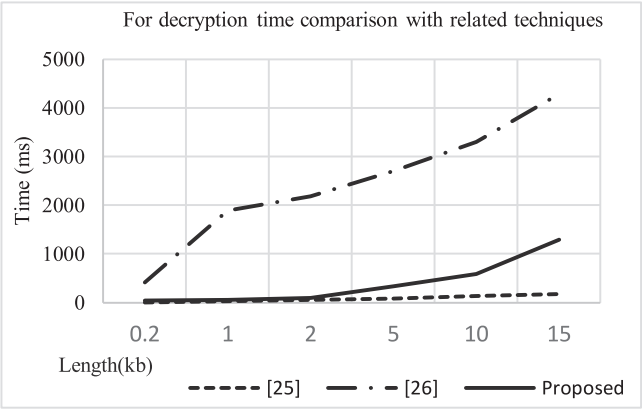


Fig. 6. Comparison in case of decryption time with related techniques.

4.7. Comparison with other techniques

Considering the time requirement of encryption and decryption processes, comparisons with techniques proposed in [25] and [26] are conferred in Figs. 5 and 6, respectively. For the proposed technique, the result of Paillier based implementation is considered. From the figure, it is seen that the technique proposed in [26] requires comparatively large time whereas the technique proposed in [25] requires comparatively less time than the proposed technique. The reason is, the technique proposed in [26] considers the symmetric key exchange, matrix operation, and XOR technique. In contrast, the technique proposed in [25] simply considers only random substitution of DNA bases with 256 ASCII characters; do not consider any other robust tool.

5. Security and statistical analyses

This section analyzes the strength and the attacks for the ciphertext produced by the proposed technique. To test the randomness of ciphertext, NIST test analyses [28] is performed.

5.1. Ciphertext strength analysis

The proposed technique embeds three levels of security while generating the ciphertext, which is as follows.

Level 1:

- 256 ASCII characters randomly can be assigned to each DNA sequence by 256! is similar to [24].
- To generate the dynamic sequence table if the sender iterates its initial form n times (where n is the number of values chosen from the mathematical series), then to breach its' secrecy the intruder also needs to repeat the process same times.

Thus, the total operations of *Level* 1 are $(256!) \times (n)$.

Level 2:

To encrypt the chunks 2048 bit key of an asymmetric cryptosystem is chosen. According to [27], in *Level* 2 to breach the secrecy of 2048 bit RSA, the intruder has to perform 2^{112} times brute force attack on the key.

Level 3:

To generate dynamic DNA encoding assuming that the sender splits the data into N chunks and the number of data within each chunk is P. Also assuming that it picks m_j DNA base sequences from NCBI bank database to merge first two chunks. Now to breach its' secrecy the number of operations that the intruder needs to perform is: $(N \times P \times m_j)$.

5.2. NIST statistical test analyses

According to the NIST statistical test suite [28], the analyses report for the proposed technique is presented in Table 6. The test consists of 10^6 bits that are obtained after processing every 100 binary sequences. For every test, the level of significance (a) is considered as 0.01. The results settle the pseudorandom characteristics of the ciphertext. Because NIST statistical test suite can consider the data only in binary form, to perform the test, the ciphertext of the proposed technique is transformed into binary. From the test, the results that are not random are marked with an asterisk.

NIST approves two ways to test the randomness of data that includes: (1) the test of the proportion of sequences passing a statistical test (1st column of Table 6), and (2) the distribution of P-values to check for uniformity (2nd column of Table 6). According to the standardization of NIST, the acceptable proportion of passing sequences should fall within the interval $0{:}99 \pm 0{:}0094392$. The proportion (3rd column of Table 6), shows that for the 'Longest Run', the 'Non-Overlapping Template' and the 'Approximate Entropy', their outputs are out of the interval. Therefore, it can be considered that the data is not completely random *i.e.* the results with an asterisk are not random. While considering the uniformity of data, the P-value is $>=0.0001$. From Table 6 it is seen that the P-value for 'Rank test', 'Universal test' and 'Approximate Entropy test' are below the standard P-value. Therefore, these three tests fail statistics uniformity test, also. However, for remaining tests, the output succeed.

5.3. Attack analysis

The strength of any cryptographic technique is related with its' sustainability over attacks. Usually to breach the secrecy of data, the intruder attempts to mount numerous possible forms of attacks. Namely, for cryptanalysis it attempts through known-plaintext attack, chosen-plaintext attack, ciphertext-only attack etc. [34–37]. Besides, to manipulate or fabricate the cipher, it tries to mount occlusion (data-loss) attack, noise attack etc. [32,33]. Thereby, the intruder tries to retrieve either some/total portion of the plaintext and/or the key. Moreover, the intruder may have the decryption key as well as a few parameters but having no information about the cipher [34].

To breach the secrecy of the proposed technique, along with the private key of the chosen asymmetric cryptosystem the intruder has to acquire many obligatory information. These are: (a) the values of dynamic sequence table, (b) two separate mathematical series, (c) the arbitrary positions picked from these series, (d) the value of dynamic DNA encoding, e) the chosen NCBI bank database etc.

Table 6
Final analyses report on NIST statistical test.

Statistical Test	*P*-Value	Proportion
Frequency	0.262249	99/100
Block Frequency	0.319084	98/100
Cumulative Sums	0.574903	99/100
Runs	0.366918	98/100
Longest Run	0.574903	100/100*
Rank	0.000017*	99/100
FFT	0.048716	98/100
Non-Overlapping Template	0.275709	95/100*
Overlapping Template	0.816537	98/100
Universal	0.000000*	99/100
Approximate Entropy	0.000000*	76/100*
Random Excursions	–	–
Random Excursions Variant	–	–
Serial	0.289667	99/100
Linear Complexity	0.085587	99/100

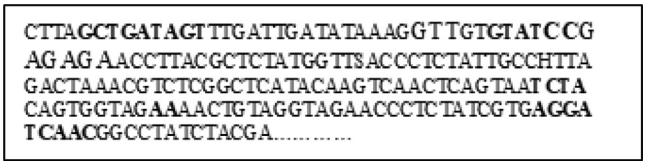


Fig. 7. 27% of final ciphertext modified by noise.

5.3.1. Known plaintext attack (KPA)

In KPA [36], the intruder has some plaintext and its matching ciphertext. Thereby, it targets to find the key corresponding to this match. Recent analyses show that many asymmetric cryptosystems are vulnerable to this attack [36]. However, to maintain its secrecy, the proposed technique relies on not only the key but also on many obligatory information and mechanisms as already specified. Hence, it is quite impossible to mount KPA for the proposed technique.

5.3.2. Chosen-plaintext attack (CPA)

In CPA, by any means the intruder obtains the matching cipher of an opted plain data [33]. To develop the proposed technique, as shown in Tables 2–4, numerous values are picked arbitrarily and randomly. Thereby while encryption; in fact, every distinct plain data adopts distinct values. Hence, the proposed technique can survive over the CPA quietly.

Similarly, it can resist against ciphertext-only attack also.

5.3.3. Noise attack

Usually noise evolves through addition and/or substitution of some random values within random positions of the cipher. Thereby while decryption, the cipher produces a deviated output. To analyze the effect of noise for the proposed technique, the same final ciphertext of Table 4 (*i.e. Step* 9) is considered. From which 27% DNA bases are modified through addition and substitution. Where the modified DNA bases are represented as bold, and a portion of the data is shown in Fig. 7.

Now through decryption the retrieved plaintext is as below:

WbZ**codmet**sAk**SK**b**E** ∼ vhT

where the bold characters are as same as the original plaintext. The difference between the original and the retrieved plaintexts is represented in Table 7. It shows that even after noising, 50% of original plaintext is accurately retrieved. Thus, the proposed technique significantly can resist against the noise attack.

5.3.4. Occlusion (data loss) attack

It may occur that a significant amount of data is cropped from anywhere of the ciphertext. To analyze the effect of occlusion for the proposed technique, the same final ciphertext (*i.e. Step* 9 of Table 4) is considered again. From the last portion of it 20% DNA bases are cropped. Now through decryption the retrieved plaintext is presented below.

WelcometoCSE, K

The difference between the original and the retrieved plaintexts is represented in Table 8. It shows that even after data cropping, 70% of original plaintext is accurately retrieved. Thus, the proposed technique significantly can resist against the occlusion attack also.

Table 7
Retrieval of 50% accurate plaintext from 27% noisy ciphertext.

Type	Text
Original Plaintext	Welcome to CSE, KUET
Retrieved Plaintext	WbZcodmetsAkSKbE∼vhT
Retrieved accurate Plaintext	W co me t S K E T

Table 8
Retrieval of 70% accurate plaintext from 20% cropped ciphertext.

Type	Text
Original Plaintext	Welcome to CSE, KUET
Retrieved Plaintext	WelcometoCSE,K
Retrieved accurate Plaintext	Welcome to CSE, K

6. Conclusions

The proposed DNA cryptographic technique introduces dynamic mechanisms to enhance the secrecy level of data. While encryption, initially the exploitation of dynamic sequence table that holds the characteristics of substitution cipher ensures first−level secrecy. Now to increase its privacy, the use of an asymmetric cryptosystem provides second−level secrecy. Finally, to boost-up its security the usage of dynamic DNA encoding that maintains the properties of product cipher offers third−level secrecy. Besides, to develop dynamic mechanisms the role of mathematical series, the iterations, the DNA base sequences of NCBI bank database etc. are also associated with its secrecy. The attack analysis, and the statistical test suite according to NIST ensures that the generated ciphertext is pseudorandom, and significantly secure. A future plan is to apply the proposed technique for various forms and platforms of data *e.g.* image, audio, numeric, video, cloud computing etc.

Declaration of Competing Interest

None.

Acknowledgment

Information & Communication Technology (ICT) Division of Ministry of Posts, Telecommunications and Information Technology, Bangladesh financially supported Md. Rafiul Biswas.

References

[1] Lu M, Lai X, Xiao G, Qin L. Symmetric-key cryptosystem with DNA technology. Sci China Ser F 2007;50(June (3)):324–33.
[2] B. Anam, K. Sakib, M. Hossain and K. Dahal, Review on the advancements of DNA cryptography, (2010). arXiv:1010.0186.
[3] Pelletier O, Weimerskirch A. Algorithmic self-assembly of DNA tiles and its application to cryptanalysis. In: Proc. of the 4th Annual Conf. on Genetic and Evolutionary Computation. Morgan Kaufmann Publishers Inc; 2002. p. 139–46.
[4] Kaliski B. Symmetric cryptosystem. Encyclopedia of cryptography and security. van Tilborg HCA, editor. Boston, MA: Springer; 2005.

[5] Galbraith SD. Mathematics of public key cryptography. Cambridge University Press; 2012.
[6] Hossain EMS, Rokibul AKM, Biswas MR, Morimoto Y. A DNA cryptographic technique based on dynamic DNA sequence table. In: Proc. of 19th IEEE int. conf. on computer and information technology (ICCIT); 2016. p. 270–5.
[7] Biswas MR, Rokibul AKM, Akber A, Morimoto Y. A DNA cryptographic technique based on dynamic DNA encoding and asymmetric cryptosystem. In: Proc. of 4th IEEE int. conf. on networking, systems and security (NSysS); 2017. p. 1–8.
[8] Borda M, Tornea O. DNA secret writing techniques. In: Proc. of 8th int. conf. on communications (COMM); 2010. p. 451–6.
[9] Zhang X, Zhou Z, Niu Y. An image encryption method based on the feistel network and dynamic DNA encoding. IEEE Photonics J 2018;10(4).
[10] Zhang Y, Liu X, Sun M. DNA based random key generation and management for OTP encryption. Biosystems 2017;159:51–63.
[11] Hirabayashi M, Nishikawa A, Tanaka F, Hagiya M, Kojima H, Oiwa K. Analysis on secure and effective applications of a DNA-based cryptosystems. In: Proc. of 6th IEEE int. conf. on bio-inspired computing: theories and applications (BICTA); 2011. p. 205–10.
[12] Khalifa A, Atito A. High-capacity DNA-based steganography. In: Proc. of 8th IEEE int. conf. on informatics and systems (INFOS); 2012. BIO-76.
[13] Fasila KA, Antony D. A multiphase cryptosystem with secure key encapsulation scheme based on principles of DNA computing. In: Proc. of 4th IEEE int. conf. on advances in computing and communications (ICACC); 2014. p. 1–4.
[14] Gao Q. A few DNA-based security techniques. In: Proc. of IEEE long island systems, applications and technology conference (LISAT); 2011. p. 1–5.
[15] Najaftorkaman M, Kazazi NS. A method to encrypt information with DNA-based cryptography. Int J Cyber-Secur Digital Forens 2015:417–26.
[16] Priya SVK, Saritha SJ. A robust technique to generate unique code DNA sequence. In: Proc. of IEEE int. conf. on energy, communication, data analytics and soft computing (ICECDS); 2017. p. 3815–20.
[17] Aieh A, Sen A, Dash SR, Dehuri S. Deoxyribonucleic acid (DNA) for a shared secret key cryptosystem with diffie hellman key sharing technique. In: Proc. of 3rd IEEE int. conf. on computer, communication, control and information technology (C3IT); 2015. p. 1–6.
[18] Reddy RPK, Nagaraju C, Subramanyam N. Text encryption through level based privacy using DNA steganography. Int J Emerg Trends Technol Comput Sci (IJETTCS) 2014;3(3).
[19] Boneh D, Dunworth C, Lipton R. Breaking DES using a molecular computer. In: Proc. of DIMACS workshop on DNA computing; 1995. p. 37–65.
[20] George JP, Kureethara JV. An efficient 2-Step DNA symmetric cryptography algorithm based on dynamic data structures. Int J Eng Technol (IJET) 2018;7(2.6):141–6.
[21] Beegom B. Jose An enhanced cryptographic model based on DNA approach. In: Proc. of IEEE Int. Conf. on Electronics, Communication and Aerospace Technology; 2017. p. 317–22.
[22] Kalsi S, Kaur H, Chang V. DNA cryptography and deep learning using genetic algorithm with NW algorithm for key generation. J Med Syst 2018;42(1).
[23] Pujari SK, Bhattacharjee G, Bhoi S. A hybridized model for image encryption through genetic algorithm and DNA sequence, 125. Elsevier, Procedia Computer Science; 2018. p. 165–71.
[24] Hammami H, Brahmi H, BenYahia S. Secured outsourcing towards a cloud computing environment based on DNA cryptography. In: Proc. of IEEE Int. conf. on information networking (ICOIN); 2018. p. 31–6.
[25] Kumar RK, Devi P. Bharathi, A novel text encryption algorithm using DNA ASCII table with a spiral approach. Int J Recent Scientific Res 2018;9(1):23588–95.
[26] Paul S. An innovative DNA cryptography technique for secure data transmission. Int J Bioinform Res Appl 2016;12(3).
[27] National Institute of Standards and Technology, Implementation guidance for FIPS PUB 140-2 and the cryptographic module validation program, updated on May 25, 2018.
[28] A.Rukhin, J.Soto, J. Nechvatal, M. Smid, and E. Barker, A statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST Special Publication 800-22 Revision 1a Revised: April 2010 Lawrence E Bassham III 2.
[29] NCBI bank genome sequence available at www.ncbi.nlm.nih.gov, accessed on August 2018.
[30] Ahmed K, El-Henawy I. Increasing robustness of data encryption standard by integrating DNA cryptography. Int J Comput Appl 2017;39(2):91–105.
[31] Enayatifar R, Guimarães FG, Siarry P. Index-based permutation-diffusion in multiple-image encryption using DNA sequence. Opt Lasers Eng 2019;115:131–40.
[32] Chen H, Tanougast C, Liu Z, Hao B. Securing color image by using hyperchaotic system in gyrator transform domains. Opt Quantum Electr 2016;48(8):396.
[33] Mondal A, Rokibul AKM, Ali GGMN, Chong PHJ, Morimoto Y. A multistage encryption technique to enhance the secrecy of image. KSII TIIS 2019;13(5):2698–717.
[34] Kumar A, Nishchal NK. Quick response code and interference-based optical asymmetric cryptosystem. JISA 2019;45:35–43.
[35] Fatima A, Mehra I, Nishchal NK. Optical image encryption using equal modulus decomposition and multiple diffractive imaging. J Opt 2016;18(8):085701.
[36] Rajput SK, Nishchal NK. Known-plaintext attack on encryption domain independent optical asymmetric cryptosystem. Opt Commun 2013;309:231–5.
[37] Fatima A, Nishchal NK. Discussion on comparative analysis and a new attack on optical asymmetric cryptosystem. JOSA A 2016;33(10):2034–40.